

Anssi Sallmén

Interaktiivinen sarjakuva internetissä

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Mediatekniikan koulutusohjelma

Insinöörityö

22.5.2015

Tekijä Otsikko	Anssi Sallmén Interaktiivinen sarjakuva internetissä
Sivumäärä Aika	30 sivua + 1 liite 22.5.2015
Tutkinto	insinööri (AMK)
Koulutusohjelma	mediatekniikka
Suuntautumisvaihtoehto	digitaalinen media
Ohjaaja	Lehtori Ilkka Kylmäniemi
<p>Insinööriyön tarkoituksena oli tehdä interaktiivinen sarjakuva internetiin. Työn tavoitteena oli perehtyä interaktiivisen sarjakuvan tekemisen eri työvaiheisiin ja -välineisiin sekä selvittää, millaisilla erilaisilla menetelmillä interaktiivisuus on sarjakuvaan mahdollista toteuttaa. Näiden tietojen pohjalta suunniteltiin ja toteutettiin interaktiivinen sarjakuva internetiin.</p> <p>Työssä perehdyttiin myös sarjakuvan historiaan ja siihen, miten se on teknologian kehityksen myötä siirtynyt digitaaliseen muotoon ja lopulta millaisia uusia muotoja se on saanut internetissä. Näitä uusia muotoja ovat muun muassa motion comic, motion novel ja tämän insinööriyön aiheena oleva interaktiivinen sarjakuva.</p> <p>Insinööriyössä sarjakuvan interaktiiviset toiminnallisuudet, kuten esimerkiksi animaatiot, äänet ja objektien liikuttamiset, toteutettiin käyttämällä mahdollisimman paljon HTML5-merkitäkieltä ja siihen tiiviisti liittyviä teknologioita, kuten esimerkiksi CSS3:a ja valmiita JavaScript-kirjastoja. Sarjakuvassa käytetty musiikki soitettiin ja äänitettiin myös itse.</p> <p>Insinööriyön tuloksena syntyi suhteellisen lyhyt ja yksinkertainen interaktiivinen sarjakuva internetiin, mutta sille asetetut toiminnallisuudet kuitenkin toimivat suunnitellusti.</p> <p>Projekti osoitti interaktiivisen sarjakuvan teon olevan varsinkin alkupään työvaiheiltaan perinteisen sarjakuvan tekemisen kaltainen, vaikkakin jo suunnittelussa tuli ottaa huomioon kohdat, joissa interaktiivisuuden aikoo toteuttaa. Projekti osoitti myös, ettei itse interaktiivisuuden toteuttaminen vaadi kovin monimutkaista koodausta. Kävi myös selväksi, että interaktiivinen sarjakuva on vielä hyvin marginaalinen sarjakuvan muoto ja jää nähtäväksi, saavuttaako se suurempaa suosiota tulevaisuudessa.</p>	
Avainsanat	sarjakuva, interaktiivisuus, digitaalinen sarjakuva, HTML5

Author Title	Anssi Sallmén Creating an interactive comic
Number of Pages Date	30 pages + 1 appendix 22 May 2015
Degree	Bachelor of Engineering
Degree Programme	Media Technology
Specialisation option	Digital Media
Instructor	Ilkka Kylmäniemi, Senior Lecturer
<p>The topic of this thesis is an interactive comic published on the Internet. The goal of the project was to become familiar with the process, tools and methods of making an interactive comic. Another goal was to find out what different methods can be used to create the interactivity of a web comic. After gathering the aforementioned information, the interactive comic was planned, created and published on the Internet.</p> <p>Part of this project was also to become familiar with the history of comics in general, including how they have evolved alongside the technological development and what kind of new forms they have gotten on the Internet. Examples of the new forms are motion comic, motion novel and interactive comic.</p> <p>In this project the interactive functionalities, like the animations, sounds and moveable objects, were created by using the HTML5 markup language and closely related technologies such as CSS3 and ready-to-use JavaScript libraries as much as possible. Music was also used in the comic, and it was self-played and recorded.</p> <p>As a result quite a short and simple interactive comic was published on the Internet. All the interactive functionalities worked as planned.</p> <p>The project showed that the making of an interactive comic is very similar to that of a traditional comic, especially the in early stages, although one must already at the planning stage think about the parts where the interactivity is going to be used. The project also showed that creating the interactivity does not require complicated programming. It also became clear that the interactive comic is still a very marginal form of comics, and it remains to be seen, if it is going to gain more popularity in the future.</p>	
Keywords	comics, interactivity, interactive comic, HTML5

Sisällys

Lyhenteet

1	Johdanto	1
2	Sarjakuvan historia	2
2.1	Perinteinen sarjakuva	2
2.2	Digitaalinen sarjakuva	4
2.3	Sarjakuva internetissä	5
3	Interaktiivinen sarjakuva internetissä	6
3.1	Interaktiivisuus lyhyesti	6
3.2	Työvaiheet ja -välineet	6
3.2.1	Ideointi ja tekstikäsikirjoitus	7
3.2.2	Luonnostelu ja kuvakäsikirjoitus	8
3.2.3	Puhtaaksi piirtäminen ja värit	9
3.2.4	Julkaiseminen	12
3.3	Interaktiivisuuden toteuttaminen	12
3.3.1	Objektin liikuttaminen	13
3.3.2	Animaatio	15
3.3.3	Ääni	17
3.3.4	Video	21
3.4	Interaktiivisuuden vaikutus sarjakuvan juoneen	23
4	Interaktiivisen sarjakuvan tulevaisuus	24
5	Yhteenveto	25
	Lähteet	27

Liitteet

Liite 1. Lähdekoodi ja CSS-tyylitiedosto

Lyhenteet ja määritelmät

HTML5	HyperText Markup Language 5. HTML-merkintäkielen viides versio.
CSS3	Cascading Style Sheets. Lähinnä WWW-dokumenteille kehitetyn tyyliohjeiston kolmas versio.
API	Application Programming Interface. Ohjelmointirajapinta.
jQuery	Avoimen lähdekoodin JavaScript-kirjasto.
jQuery UI	jQuery User Interface. Käyttöliittymäkomponentteihin keskittynyt avoimen lähdekoodin JavaScript-kirjasto.
WAV	Waveform Audio File Format. Äänen tallentamiseen käytetty tiedostomuoto.
MP3	MPEG-1 Audio Layer 3. Häviöllinen äänenpakkausjärjestelmä.
MP4	MPEG-4 Part 14. Säiliömuoto, joka voi sisältää videokuvaa, ääntä sekä tekstityksen.

1 Johdanto

Insinööriyön tavoitteena on perehtyä interaktiivisen sarjakuvan tekemisen eri työvaiheisiin ja -välineisiin ja lopuksi suunnitella ja toteuttaa interaktiivinen sarjakuva internetiin käyttämällä HTML5-merkintäkieltä ja siihen tiiviisti liittyviä teknologioita, kuten CSS3:a ja JavaScript-kirjastoja. Tavoitteena on myös tutkia, millaisilla erilaisilla keinoilla ja tekniikoilla interaktiivisuus voidaan toteuttaa internetsarjakuvassa. Insinööriyötä ei tehdä millekään yritykselle, vaan se on enemmänkin kokeiluluontoinen.

Koska minulla ei ole aiempaa kokemusta sarjakuvan teosta tai tietämystä sen historiasta, perehdyn myös lyhyesti siihen, miten moderni sarjakuva on saanut alkunsa ja kehittynyt niin kutsutusta perinteisestä, paperilla julkaistavasta sarjakuvasta erilaisiin digitaalisiin muotoihin, joksi myös tämän työn aiheena oleva interaktiivinen sarjakuva lasketaan. Itse aiheen valitseminen oli suhteellisen helppoa, sillä vaikka en olekaan piirtänyt sarjakuvia, olen aina pitänyt piirtämisestä, joten tässä työssä yhdistyvät sopivalla tavalla taiteellisuus, luovuus ja mediatekniikan teknisempi puoli.

2 Sarjakuvan historia

2.1 Perinteinen sarjakuva

Sarjakuvalle ei ole olemassa absoluuttisen tarkkaa määritelmää, mutta suomenkielinen termi, sarjakuva, on yllättävänkin osuva (vrt. englanninkielinen termi comic), sillä muun muassa amerikkalainen sarjakuvantekijä ja kirjailija Scott McCloud määrittelee sarjakuvan kirjassaan *Reinventing Comics* sarjaksi rinnakkaisia kuvia, joilla on harkittu järjestys (1, s. 200).

Sarjakuvamaisella kerronnalla on pitkät juuret, ja sen tarkkaa alkupistettä on mahdotonta määrittää. Sarjakuvamaista kerrontaa edustavat esimerkiksi muinaisen Egyptin seinämaalaukset, Trajanuksen pylvään reliefit ja keskiaikaiset kuvakudokset, joissa kaikissa kuvataan erilaisia tapahtumia pitkinä kaistaleina (1, s. 217–218).

Edellä mainittujen esimerkkien sarjakuvamaisesta tyylistä huolimatta nykymuotoisen, ja perinteisen, paperille painetun sarjakuvan katsotaan kuitenkin saaneen alkunsa vasta 1890-luvulla amerikkalaisissa sanomalehdissä. Yhtenä ensimmäisistä nykyaikaisista sarjakuvista pidetään kuvassa 1 esiintyvää, Richard Outcaultin 1890-luvun loppupuolella luomaa ja ensiesiintymisensä newyorkilaisessa sanomalehdessä tehnyttä *The Yellow Kidiä* (suom. Keltainen kakara). Outcault kokeili ja käytti sarjakuvissaan monia elementtejä, muun muassa puhekuplia ja päähahmoa, jonka edesottamuksia seurataan sarjakuvasta toiseen, joita yhä nykyäänkin pidetään oleellisena osana sarjakuvaa. (2, s. 1–15.) Mitä suomalaisen sarjakuvan historiaan tulee, ensimmäinen suomalainen sarjakuva-albumi, Professori Itikaisen tutkimusretki, julkaistiin reilu sata vuotta sitten, vuonna 1911 (3, s. 13–14).



Kuva 1. Richard Outcaultin piirtämä Yellow Kid -sarjakuvastriippi vuodelta 1896 (4).

Nämä alkuaikojen humoristiset ja usein jopa pilakuvamaiset sanomalehtistriipit kasvattivat tasaiseen tahtiin suosiotaan, ja 1900-luvun alkupuolella niistä ruvettiin koostamaan myös sarjakuva-albumeita. Vähitellen 1920- ja 1930-lukujen aikana sarjakuva alkoi muuttua. Albumien sisältö vaihtui koosteista yksittäisiin varta vasten niitä tehtyihin pitempiin tarinoihin, ja myös sarjakuvien aiheet muuttuivat samoihin aikoihin. Humorististen aiheiden rinnalle alkoi nousta muun muassa toiminta- ja supersankariaiheisia sarjakuvia. (5; 6.) Amerikkalaisista sarjakuvista hyvinä esimerkkeinä tästä muutoksesta voidaan pitää 1930-luvun taitteessa ensiesiintymisensä sarjakuvassa tehnyttä Tarzania ja vuonna 1938 ensiesiintymisensä tehnyttä Supermaniä (suom. Teräsmies) (2, s. 221–224, 256). Euroopassa ehkä tunnetuin tapaus tästä muutoksesta on belgialaisen sarjakuvataiteilija Hergén 1920-luvun lopussa luoma reportteri Tintin (suom. Tintti) (7, s. 8).

1940-luvulla suosituin sarjakuvan muoto, varsinkin Yhdysvalloissa, oli muun muassa Teräsmiehen myötä syntynyt supersankarisarjakuva. Supersankarisarjakuvien suosion alamäki alkoi kuitenkin jo toisen maailmansodan jälkeen ja jatkui tasaiseen tahtiin 1950-luvun alkupuolelle. Tätä lyhyttä aikakautta kutsutaan nimellä The Golden Age of Comics (suom. sarjakuvien kultakausi). Kuitenkin muutaman hiljaisen vuoden jälkeen

ne kokivat jo saman vuosikymmenen loppupuolella uuden tulemisen, ja siitä lähtien niiden suosio on pysynyt suurena aina näihin päiviin saakka. (8.)

Erilaisista uusista piirrostyleistä ja aihealueista huolimatta paperille painettu sarjakuva on pysynyt toteutukseltaan hyvinkin samanlaisena 1930-luvulta lähtien. Edelleen käytetään samoja hyväksi havaittuja keinoja, kuten puhekuplia ja ruutuja. Tähän syynä on tietysti julkaisualustana toimiva staattinen paperi, ja sen tuomat rajoitukset, kuten arkin koko ja niin edelleen. Muutoksen tähän toi digitaalinen sarjakuva, joka syntyi paperisen sarjakuvan rinnalle tietokoneiden vähitellen kehittyessä.

2.2 Digitaalinen sarjakuva

Digitaalisella sarjakuvalla voidaan tarkoittaa joko kokonaan digitaalisesti toteutettua sarjakuvaa tai digitaalisesti julkaistua sarjakuvaa. Ensimmäiset digitaaliset sarjakuvat ilmestyivät 1980-luvulla, ja ensimmäinen kokonaan digitaalisesti tuotettu sarjakuva oli vuonna 1985 ilmestynyt Shatter. Sen kuvat oli piirretty Applen Macintosh-tietokoneella. Shatter ainoastaan piirrettiin tietokoneella, sitä ei julkaistu digitaalisessa muodossa. (1, s. 140.) Tietokoneen avulla piirtäminen on piirto- ja kuvankäsittelyohjelmien ja piirtopöytien kehittymisen myötä yleistynyt, mikä ei sinällään yllätä, sillä piirtopöydän ja tietokoneen käytössä on kiistattomia etuja paperille piirtämiseen verrattuna. Esimerkiksi tietokoneella piirrettäessä ei ole vaaraa paperin rypistymisestä, tahraantumisesta tai repeämisestä. Objekteja voi piirtää eri tasoille, ja niitä voi liikutella ja ”kumittaa” vapaasti ja niin edelleen (9).

Digitaalinen sarjakuva kehittyi käsi kädessä teknologian kehityksen kanssa, ja jo 1990-luvun alkuvuosina sarjakuvia toteutettiin muun muassa vektorigrafiikkaa ja 3D-mallinnusta hyväksi käyttäen. Hyvänä esimerkkinä tästä on Pepe Morenon toteuttama Batman: Digital Justice, jonka teossa hyödynnettiin 3D-mallinnusta, vektorigrafiikkaa ja CAD-ohjelmia (Computer-Aided Design eli tietokoneavusteinen suunnittelu) (10). Myös aikansa huipputeknologiaa edustanutta CD-ROMia käytettiin sarjakuvien julkaisualustana, niiden suosio tosin jäi vähäiseksi (1, s. 208).

Digitaalisessa muodossa julkaistavat sarjakuvat antavat tekijöille huomattavasti vapaammat kädet sarjakuvan toteutuksen suhteen verrattuna paperiseen sarjakuvaan. Paperille painettavassa sarjakuvassa on aina rajana sivun tai aukeaman koko, kun

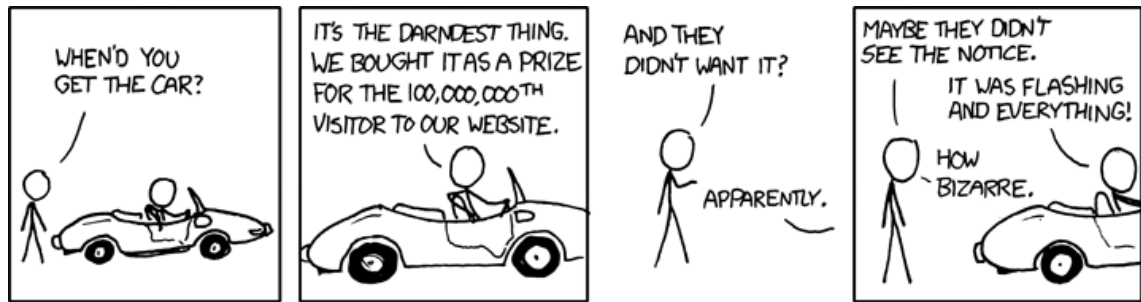
taas tietokoneen näyttö on käytännössä rajaton. Tietokoneiden mukaantulo on mahdollistanut myös äänen, animaation ja videon käyttöönoton. Näiden teknologian suomien vapauksien ansiosta alkoikin syntyä erilaisia kokeellisia sarjakuvan muotoja, joista yksi esimerkki on, tämän insinööriyön aiheenakin oleva, interaktiivinen sarjakuva. (1, s. 222.)

Toinen, interaktiivisen sarjakuvan ohella, digitaalisen sarjakuvan uusimmista muodoista on motion comic. Se on nimensä mukaisesti liikkeessä, ja se eroaakin perinteistä sarjakuvasta siten, että siinä lukija ei voi määrittää lukutahtia, vaan sarjakuva rullaa animaation tavoin eteenpäin. Lyhyesti sanottuna motion comic on perinteisen sarjakuvan ja animaation yhdistelmä. Usein myös puhekuplat on korvattu puheella ja ääniefekteillä. Yhtenä ensimmäisistä pidetään vuonna 2005 ilmestynyttä *Saw: Rebirthiä*. Motion comicin pohjalta on myöhemmin syntynyt myös motion novel. Se eroaa motion comicista siten, että se yhdistelee vielä vapaammin tietokonegrafiikkaa, animaatiota, ääntä ja valokuvia perinteiseen sarjakuvankerrontaan. (11, s. 15–19.)

2.3 Sarjakuva internetissä

Internetsarjakuvalla tarkoitetaan internetissä julkaistavaa sarjakuvaa. Ensimmäiset internetsarjakuvat julkaistiin jo 1980-luvun loppupuoliskolla hyödyntämällä CompuServen ja Q-Linkin kaltaisia varhaisia verkkopalveluita. Ensimmäisinä pidetään Eric Millikin vuonna 1985 julkaisemaa Ihmemaa Ozia parodioivaa *Witches and Stitches* -sarjakuvaa sekä Joe Ekaitisin vuotta myöhemmin aloittamaa *T.H.E. Foxia*. (12, s. 1; 13.)

Edellä mainituista uusista muodoista ja lähes rajattomista mahdollisuuksistaan huolimatta digitaalisessa muodossa julkaistu sarjakuva ei toistaiseksi ole saavuttanut, muutamaa poikkeusta lukuun ottamatta, kovinkaan suurta suosiota. Yksi esimerkki maailmalla suositusta internetsarjakuvasta on Randall Munroen luoma *xkcd*. Se saavutti jo muutamassa vuodessa aloittamisestaan 80 miljoonaa sivunkatsomiskertaa kuukaudessa. (14.) Kuten kuvassa 2 näkyy, se, kuten moni muukin internetissä julkaistu sarjakuva, noudattaa silti yhä perinteistä sanomalehdistä tuttua keskimäärin kolmen ruudun formaattia. Kotimaiseksi esimerkiksi tällaisesta sarjakuvasta voisi mainita muun muassa Helsingin Sanomien internetsivuilla päivittäin ilmestyvän *Fingerporin* (15).



Kuva 2. Esimerkkistriippi Randall Munroen piirtämästä xkcd-sarjakuvasta (16).

Kokeellisempien digitaalisen sarjakuvan muotojen, kuten interaktiivisen sarjakuvan ja motion comicin, suosio näyttää jääneen vähäisemmäksi, mikä kävi ilmi muun muassa tämän insinööriyön tiedonhakuaiheessa vähäisinä osumina googlen hakutuloksissa.

3 Interaktiivinen sarjakuva internetissä

3.1 Interaktiivisuus lyhyesti

Interaktiivisuudella tarkoitetaan vuorovaikutteista viestintää. Vuorovaikutteisessa viestinnässä käyttäjällä on mahdollisuus antaa palautetta ja annetun palautteen avulla osallistua ja vaikuttaa viestintävälineen sisältöön (17), esimerkkinä vaikkapa televisio-ohjelmassa järjestettävä yleisöäänestys, jossa yleisö äänestämällä vaikuttaa ohjelman kulkuun. Seuraavissa luvuissa käsittelen tarkemmin, millaisilla erilaisilla tavoilla ja tekniikoilla interaktiivisuus on mahdollista toteuttaa internetsarjakuvassa.

3.2 Työvaiheet ja -välineet

Sarjakuvanteon työvaiheet koostuvat karkeasti seuraavista neljästä vaiheesta:

- 1) yllyke
- 2) suunnitteleminen
- 3) toteuttaminen

4) julkaiseminen.

Yllykkeenä voi toimia hyvinkin moni eri asia, esimerkiksi tarve kertoa tarina tai pelkkä hyvän idean keksiminen (18). Suunnitteluvaihe taas sisältää muun muassa luonnostelun ja käsikirjoitukset. Toteuttamisvaiheessa sarjakuva työstetään valmiiksi luonnosten ja käsikirjoitusten pohjalta, ja lopuksi valmis sarjakuva julkaistaan yhdessä tai useammassa halutussa julkaisuformaatisissa. (19.)

Mainitut neljä vaihetta ovat suurelta osin samanlaisia niin interaktiivisessa internetsarjakuvassa kuin perinteisessä paperisarjakuvassakin. Suurimmat erot interaktiivisen internetsarjakuvan ja perinteisen sarjakuvan välillä ovat loppupään tietokoneella toteutettavat työvaiheet, joissa itse interaktiivisuus toteutetaan, sekä tarpeen mukaan toteutetut musiikin äänitys ja videoiden kuvaus.

Käytettävät työvälineet taas riippuvat pitkälti siitä, kuinka paljon tekijä haluaa hyödyntää tietotekniikkaa sarjakuvan teossa. Projektina interaktiivisen sarjakuvan pystyy halutessaan alun ideoinnista ja luonnostelusta aina väritysvaiheeseen asti toteuttamaan hyvinkin pitkälle pelkällä paperilla ja kynillä. Tässä projektissa puhtaaksi piirtäminen ja väritys toteutettiin käyttämällä tietokoneeseen yhdistettyä piirtopöytää. Sarjakuvan ruutujen puhtaaksi piirtäminen ja väritys tehtiin Adobe Photoshopilla.

Mitä taas tulee interaktiivisessa sarjakuvassa mahdollisesti käytettävien musiikin, äänten ja videoiden tuottamiseen, niiden tekoon voidaan käyttää moniakin erilaisia ja eritasoisia työvälineitä. Käytettävät laitteet riippuvat pitkälti siitä, kuinka hyvälaatuista kuvaa ja ääntä sarjakuva ja/tai sen tekijä vaatii. Yksi suhteellisen helppo tapa, jota tässäkin työssä käytettiin, on hoitaa äänten ja videon tallennus videota ja ääntä tallentavalla digikameralla. Äänten editointiin tietokoneella käytettiin Abletonin Live 8 -ohjelmaa.

3.2.1 Ideointi ja tekstikäsikirjoitus

Kuten todettua, alkupään työvaiheiltaan ja -välineiltään interaktiivisen internetsarjakuvan tekeminen on hyvin samankaltaista kuin perinteisenkin sarjakuvan. Interaktiivisen sarjakuvan teko alkaa aivan kuten minkä tahansa muunkin tarinan tekeminen. Ensin valitaan idea, jota lähdetään toteuttamaan. Kun toteutuskelpoinen

idea on valittu, pitää seuraavaksi suunnitella tarinan juoni, päähenkilöt, tapahtuma-aika ja -paikka ja niin edelleen. (20.) Suurimpana erona interaktiivisen ja perinteisen sarjakuvan ideoinnissa on tietysti se, että interaktiivisen sarjakuvan tekijän tulee jo ideointivaiheessa miettiä kohdat, joissa aikoo käyttää vuorovaikutusta, esimerkiksi missä kohtaa ja miten lukija voi vaikuttaa juonen kulkuun.

Kun ideointi on saatu valmiiksi, on seuraava vaihe useimmiten tekstikäsikirjoitus. Sille ei ole olemassa mitään yhtä oikeata mallia, mutta tietty tehtävä sillä kyllä on. Sen tehtävänä on kuvata sanoin, miten valittu tarina kerrotaan, eli kuvata, mitä ruuduissa tapahtuu, minkälainen tausta niissä on ja niin edelleen (21). Tekstikäsikirjoitus ei tämän insinööriyön sarjakuvassa ole mitenkään kovin suuressa roolissa, sillä sarjakuvan runkona toimii olemassa olevaa tarina. Sarjakuvan juoni pohjautuu 1900-luvun alkupuoliskolla eläneistä amerikkalaisista blues-kitaristeista kertoviin tarinoihin. Legendan mukaan muun muassa kitaristit Tommy Johnson ja Robert Johnson olisivat saaneet kitaransoittonsa myymällä sielunsa paholaiselle tienristeyksessä keskellä yötä. (22; 23.)

3.2.2 Luonnostelu ja kuvakäsikirjoitus

Luonnoksilla tarkoitetaan usein nopeasti ja karkeasti piirrettyjä hahmotelmia, joilla pyritään saamaan ideoita konkreettiseen muotoon (24). Sarjakuvissa, olipa sitten kyse tavallisesta tai interaktiivisesta sarjakuvasta, tällä tarkoitetaan esimerkiksi hahmojen ja tapahtumapaikkojen hahmottelua. Luonnosten ei tarvitse, eikä pidäkään, olla lopullisia, vaan niiden avulla vain luodaan pohjaa hahmoille ja tapahtumapaikoille, jotka sitten seuraavassa vaiheessa piirretään puhtaaksi. Luonnostelun voi tehdä joko käsin piirtämällä paperille tai piirtopöydän ja tietokoneen avulla. Tämän insinööriyön luonnosteluvaihe toteutettiin lyijykynällä ja paperilla.

Hahmojen tai tapahtumapaikkojen luonnostelulle ei ole olemassa mitään absoluuttista järjestystä tai määrää, vaan tekijällä on varsin vapaat kädet niiden suhteen. Yksi looginen tapa aloittaa luonnostelu on aloittaa se tarinan päähenkilöistä, joita tässä tapauksessa on vain yksi, koska hän on mukana lähes jokaisessa sarjakuvan ruudussa. Siksi hänen ilmeensä, kasvonpiirteensä, ruumiinrakenteensa sekä erilaiset asennot ja kuvakulmat on hyvä hahmotella riittävän tarkkaan jo luonnosteluvaiheessa. Päähenkilön lisäksi on suositeltavaa luonnostella sarjakuvan sivuhenkilöt ja sarjakuvan tapahtumapaikat. Inspiraatiota tämän insinööriyön henkilöille ja tapahtumapaikoille

haettiin Googlen kuvahaun avulla vanhoista 1900-luvun alkupuolen valokuvista ja blueslevyjen mainoksista.

Kuvakäsikirjoituksen tarkoituksena taas on kuvata tarinan juoni kuvina. Sillä haetaan kuviin, ja interaktiivisen sarjakuvan kohdalla myös kohtauksiin, halutut sommitelmat, kuvakulmat ja toiminnan suunta (25). Kuvassa 3 on esimerkki varhaisesta kuvakäsikirjoituksesta. Kuten kuvaa 3 ja valmista työtä vertaamalla huomataan, ei kuvakäsikirjoitus ole mitenkään lopullinen versio, vaan sen tarkoitus on ainoastaan helpottaa lopullisen työn tekoa.



Kuva 3. Varhainen kuvakäsikirjoitus.

3.2.3 Puhtaaksi piirtäminen ja värit

Puhtaaksi piirtämisellä tarkoitetaan luonnosten puhtaaksi piirtämistä, eli piirretään kuvalle lopulliset ja selkeät ääriviivat. Puhtaaksi piirtämisestä käytetään usein myös nimitystä tussaus. Ennen tietokoneiden käytön yleistymistä perinteinen tapa toteuttaa tussaus oli sanan mukaisesti tussilla, eli asetetaan luonnos valopöydälle ja sen päälle tyhjä paperi, johon mustalla tussilla piirretään luonnosta mukaillen lopulliset viivat.

Internetsarjakuvan ollessa kyseessä täytyy nämä valmiit viivapiirroksot vielä skannata tietokoneelle. Tussauksen pystyy toki toteuttamaan myös ilman valopöytä. Siinä tapauksessa ääriiivat piirretään suoraan lyijykynäluonnoksen päälle, minkä jälkeen kumitetaan ylimääräiset lyijykynän jäljet pois. Tämä tekniikka ei kuitenkaan ole paras mahdollinen, sillä kumittamisesta jää aina jälkiä, joiden poisto aiheuttaa turhaa työtä tietokoneelle skannauksen jälkeen. (26.)

Toinen, nykyaikaisempi, vaihtoehto, jota tässäkin työssä käytettiin, on tehdä ääriiivojen puhtaaksipiirtäminen kokonaan tietokoneella. Aluksi luonnokset skannataan tietokoneelle. Skannausasetuksista sen verran, että värimuodolla, olipa se sitten mustavalkoinen, harmaasävy tai värikuva, ei ole tässä tapauksessa merkitystä, sillä skannattu luonnos toimii kuvankäsittelyohjelmassa ainoastaan taustakuvana, jonka päälle luodaan viivapiirrosta varten oma taso. Samasta syystä myöskään valittavalla tiedostomuodolla ei ole suurta merkitystä. Pikselitiheydeksi eli tarkkuudeksi on hyvä valita vähintään 300 ppi (pixels per inch eli pikseleiden määrä tuumaa kohti).

Viimeistään kuitenkin tässä vaiheessa, ennen kuvien valmiiksi piirtämistä ja värittämistä tietokoneella, on myös hyvä miettiä julkaistavan sarjakuvan kuvien ja sivujen kokoa ja muotoa. Kuten luvussa 2.2 jo todettiin, on tietokoneen näyttö käytännössä rajaton kuvan koon suhteen, joten se mahdollistaa hyvinkin erilaisia tapoja näyttää sarjakuvaa. Sarjakuva voi siis esimerkiksi olla yksi suuri kuva, jota käyttäjä sitten voi rullata auki omaan tahtiin, tai se voi olla perinteisen paperisarjakuvan sivuja ja aukeamia imitoiva. Tähän työhön valittiin kuitenkin yksi sarjakuvaruutu yhtä sivua kohti. Täten, kun lukija ei näe seuraavaa ruutua, hänellä säilyy paremmin jännitys ja tietty epätietoisuus tulevista tapahtumista.

Tietokoneiden näytöt ovat suhteessa leveämpiä kuin korkeita, ja siksi vaakasivut toimivat usein pystysivuja paremmin (27). Koska tämän insinöörityön sarjakuvassa kaikki ruudut käyttävät samaa, valokuva-albumin kaltaista pohjasivua, ei itse ruutujen muodolla ollut niin väliä, vaan ne muokattiin aina sopimaan pohjakuvan raameihin.

Ruutujen luonnostelu aloitettiin paperilla. Sen jälkeen valmiit luonnokset skannattiin tietokoneelle, jossa ne avattiin Adoben Photoshopilla. Photoshopissa työt aloitettiin luomalla uusi tyhjä taso (layer) avatun luonnoskuvan päälle. Tähän tasoon sitten ”tussataan” mustalla ääriiivat. Tussauksen jälkeen luodaan ääriiivatason alapuolelle lisää tasoa väritystä varten, jotta ääriiivat jäävät näin päällimmäisenä näkyviin.

Väriyksellä tarkoitetaan sarjakuvan ruutujen väritystä. Sen toteutukseen pätevät samat vaihtoehdot kuin tussauksenkin suhteen, eli väriksen voi hoitaa käsin jo paperivaiheessa tai sitten vasta tietokoneella. (28.) Tietysti vielä yhtenä vaihtoehtona on yhdistellä molempia tekniikoita eli esimerkiksi tussata kuvat käsin ja värittää ne vasta tietokoneella.

Väritystasoja käytettiin työssä siten, että alimalla tasolla oli taustaväri, jonka päälle väritettiin kerros kerrokselta eri pintoja. Kuvassa 4 näkyvässä esimerkissä sininen tausta on alimpana omana tasonaan, sen päälle väritettiin puut omalle tasolleen, jonka päälle talo omalle tasolleen ja niin edelleen. Koska paperille piirretyt kuvat olivat luonnosmaisia, ei aivan jokaista kuvaa aivan yksi yhteen tussattu ja väritetty, vaan kuvia muokattiin ja niihin lisättiin erilaisia yksityiskohtia tarpeen mukaan vielä tietokoneella. Piirtämisen ja värittämisen työvälineenä käytettiin Wacomin Intuos4-piirtopöytää, joka tietokoneen hiiren sijasta sopii huomattavasti paremmin tavallisilla kynillä työskentelyyn tottuneelle piirtäjälle.



Kuva 4. Piirrosvaiheet. Vasemmalla luonnos, keskellä ääri viivat ja oikealla valmis kuva.

Tussaus- ja väritysvaiheiden tarve ja laajuus riippuvat hyvin paljon esimerkiksi tekijästä ja valitun sarjakuvan tyylistä. On täysin mahdollista piirtää sarjakuva suoraan lopulliseen muotoonsa ilman luonnosvaihetta, eli tällöin ei luonnosten niin sanottua tussausta tarvita. Myöskään sarjakuvan väritys ei ole pakollista, sillä monet sarjakuvat ovat mustavalkoisia.

3.2.4 Julkaiseminen

Perinteisen staattisen sarjakuvan pystyy julkaisemaan painettuna monessakin eri formaatissa, sekä digitaalisissa muodoissa internetissä, CD-levylle poltettuna tai pelkkänä yksittäisenä tiedostona. Interaktiivisen sarjakuvan kohdalla taas interaktiivisuuden luomiseen käytetyt äänet, animaatiot, videot ja niin edelleen rajaavat ulos nykyisin käytössä olevat painetut formaatit. Interaktiivisen sarjakuvan pystyy siis julkaisemaan ainoastaan digitaalisessa muodossa.

3.3 Interaktiivisuuden toteuttaminen

Interaktiivisuuden internetsarjakuvaan pystyy toteuttamaan erilaisilla menetelmillä, esimerkiksi Adobe Flashin tai JavaScriptin avulla. Tämän insinöörityön sarjakuva on kuitenkin toteutettu suurimmaksi osaksi HTML5:n avulla. Tarkoituksena oli kokeilla HTML-merkintäkielen mukanaan tuomia uusia elementtejä ja ohjelmointirajapintoja (engl. Application Programming Interface eli API). Näitä uusia elementtejä ovat muun muassa <audio>- ja <video>-elementit, jotka nimensä mukaisesti mahdollistavat ääni- ja videotiedostojen käytön ja täten sopivat täydellisesti interaktiivisen sarjakuvan luomiseen. Eräs API, jota hyödynsin, on JavaScriptin avulla toimiva niin sanottu drag-and-drop- eli raahaa ja pudota -toiminto.

HTML5:n mukanaan tuomista uudistuksista huolimatta sillä ei pysty aivan kaikenlaista interaktiivisuutta toteuttamaan, joten osa toiminnallisuuksista toteutettiin käyttämällä hyväksi JavaScriptiä, jQueryä, joka on avoimen lähdekoodin JavaScript-kirjasto, sekä jQuery:n päälle rakennettua jQuery UI:tä, johon on kerätty yhteen muun muassa erilaisia käyttöliittymäkomponentteja, efektejä ja teemoja. Avoin lähdekoodi siis tarkoittaa sitä, että koodi on käyttäjän vapaasti tutkittavissa, käytettävissä, muokattavissa, kopioitavissa ja jaettavissa (29). JavaScript-kirjastolla taas tarkoitetaan

valmiiksi kirjoitettua JavaScript-koodia, jonka osia käyttäjä voi suhteellisen yksinkertaisesti hyödyntää joutumatta itse kirjoittamaan rivikaupalla samaa koodia (30).

Työssä hyödynnettiin myös HTML5-merkintäkieleen tiiviisti liittyvää CSS3:ta ja sen mukanaan tuomia uusia ominaisuuksia. CSS (Cascading Style Sheets), eli porrastetut tyyliarkit, on www-dokumentteja varten kehitetty tyyliohjeiden säännöstö, jolla määritetään, miltä dokumentti näyttää selaimessa (31). Näistä CSS3:n uusista ominaisuuksista ylivoimaisesti tärkein ja oikeastaan ratkaiseva koko projektin toteuttamisen suhteen oli CSS3:n animaatio-ominaisuus. Tästä animaatio-ominaisuudesta kerron tarkemmin luvussa 4.2.2.

3.3.1 Objektin liikuttaminen

Mahdollisuus liikuttaa objektia sarjakuvassa on yksi selkeimmistä tavoista toteuttaa vuorovaikutus sarjakuvan ja lukijan välillä. Objektin liikuttamisen voi tehdä usealla erilaisella menetelmällä. Työssä toteutettiin objektien liikkuttaminen kahdella eri tavalla ja menetelmällä. Sarjakuvassa ensimmäisenä vastaantulevan lukijan liikuteltavissa olevan objektin, tyhjän pullon, liikuttaminen toteutettiin jQuery UI:n avulla. Toinen työssä käytetty menetelmä käyttää hyväkseen edellä mainittua HTML5:n drag-and-drop API:a. Mainittakoon vielä esimerkkinä, että hahmon ohjaaminen nuolinäppäimillä, mitä ei tässä työssä käytetty, olisi myös yksi tapa hoitaa objektin liikuttaminen.

Objektin liikuttaminen jQuery UI:n avulla on suhteellisen yksinkertaista. Ensin tarvitsee ladata internetistä valmiit jQuery- ja jQuery UI -JavaScript-kirjastot, jotka ovat yksittäisiä isoja JavaScript-tiedostoja ja joihin pitää linkata `<script>`-elementillä haluttu sivu HTML-koodissa. Tämän alkuvalmistelun jälkeen määritettiin `<div>`-elementillä alue, jonka sisällä objektia pystyy ainoastaan liikuttamaan, jottei kyseistä tyhjää pulloa voi raahata ulos sarjakuvan ruudusta. Liikuteltava pullo on tässä tapauksessa oman `<div>`-elementin sisällä oleva kuvatiedosto. Tämän jälkeen tarvitaan enää itse liikuttamisfunktio, joka sekin on hyvin yksinkertainen. Tarvitaan `<script>`-elementti, jonka sisään luodaan funktio, jossa ensin valitaan haluttu HTML-elementti, joka tässä tapauksessa on `draggable`-niminen `<div>`-elementti. Sen perään liitetään vielä toiminta, joka määrittää, mitä elementille tehdään. (32.) Esimerkkikoodissa 1 on pullon liikuttamiseen vaadittava koodi kommentoituna.

```

<!DOCTYPE html>
<html>
<head>
<title>Crossroads</title>
<meta charset="UTF-8">
<link rel="stylesheet" type="text/css" href="css/leiska.css">
<!--linkkaukset jQuery ja jQuery UI tiedostoihin-->
<script src="js/jquery-1.11.1.min.js"></script>
<script src="js/jquery-ui.min.js"></script>

<!--alla liikuttamisfunktio, jossa on div-elementistä, nimeltä
draggable, tehdään liikuteltava ulomman div-elementin sisällä-->
<script>
$(function() {
$( '#draggable' ).draggable({ containment: "parent"});
});
</script>
</head>
<body>
<div id="wrapper">

<a href="../inssi/page3"><div id="back"></div></a>
<a href="../inssi/page5"><div id="button"></div></a>
<!--Rajattu alue, jossa objektia pystyy liikuttamaan-->
<div id="dragarea">
<!--itse liikuteltava objekti-->
<div id="draggable"></div>
</div>
</div>
<div id="footer"></div>
</body>
</html>

```

Esimerkkikoodi 1. Objektin liikuttaminen jQuery UI:n avulla.

HTML5:n drag-and-drop API:lla taas toteutettiin kohta, jossa lukijan tulee raahata ja pudottaa haluamansa vastausvaihtoehto tiettyyn kohtaan sivua. Tämä aloitettiin luomalla <div>-elementillä alue, johon rahaattu vaihtoehto pudotetaan. Tähän <div>-elementtiin pitää lisätä ondrop- ja ondragover-määrittäykset. Näistä kahdesta ensimmäinen määrittää, mikä JavaScript-funktio suoritetaan, kun alueelle pudotetaan jotakin. Ondragover-määrittäminen taas määrittää, mikä JavaScript-funktio suoritetaan, kun elementti raahataan alueen päälle. (33.)

Raahattavat elementit, jotka tässä tapauksessa ovat kuvia, vaativat myös kaksi lisämäärittystä nimeltään draggable ja ondragstart. Jotta kuva olisi raahattavissa, pitää draggable-määrittämiselle laittaa arvoksi true (eli tosi). Ondragstart-attribuutilla määritetään, mikä JavaScript-funktio suoritetaan, kun elementin raahaus aloitetaan. Molemmille raahattaville kuville pitää luonnollisesti määrittää omat raahausfunktiot, sillä riippuen siitä, kumman kuvan raahaa ja pudottaa, avautuu sarjakuvasta eri sivu. (33.)

Itse JavaScript-funktioita tarvitaan kolme erilaista: allowDrop, drag ja drop. Näistä ensimmäisellä, nimensä mukaisesti, hyväksytään elementtien pudottaminen sivulla, mikä siis oletusarvoisesti on estetty. Drag-funktiolla määritetään, minkälaisista dataa raahattavan elementin mukana kuljetetaan. Tässä tapauksessa siinä kuljetetaan mukana linkki seuraavalle sivulle. Viimeisenä tarvitaan vielä drop-funktio, jolla avataan drag-funktiolla siirretty linkki. (33.) Tämän drag-and-drop API:lla toteutetun kohdan täydellinen HTML-koodi funktioineen löytyy liitteen 1 sisältämästä lähdekoodista.

3.3.2 Animaatio

Tässä insinööriyössä käytetyt animaatiot ovat itse tehtyjä ja koodattuja, mutta nykyään on olemassa myös valmiita sovelluksia, jotka hyödyntävät HTML5:tä, CSS3:a ja JavaScriptia ja joiden käyttämän graafisen käyttöliittymän avulla animaatioiden teko on helppoa. Esimerkkeinä tällaisista sovelluksista mainittakoon vaikkapa Adoben julkaisema Edge Animate ja Googlen vastaavankaltainen sovellus Google Web Developer. (34; 35.)

Edellä mainitulla CSS3:n sisältämällä animaatio-ominaisuudella käyttäjä pystyy korvaamaan muun muassa animoidut kuvat, Flash-animaatiot ja JavaScriptillä toteutetut animaatiot. CSS3-animaatio on tietyn elementin tyylin muuttamista asteittain yhdestä tyylistä toiseen. Näitä asteita voi olla niin monta, kuin halutaan. (36.)

Esimerkkikoodissa 2 on yksinkertainen ja varsinaiseen työhön kuulumaton esimerkki CSS3-animaatiosta. Siinä animoidaan 50 x 50 pikselin kokoinen sininen neliö neljässä sekunnissa vaiheittain läpinäkyväksi.

```
<!DOCTYPE html>
<html>
<head>
<style>
div{ width: 50px;
height 50px;
background-color: blue;
/* animaation nimi sekä kesto */
animation: animaatio1 4s;}
@keyframes animaatio1 {
/* animaation asteet/vaiheet joissa läpinäkyvyys kasvaa */
25% {opacity: 0.75;}
50% {opacity: 0.5;}
100% {opacity: 0.0;}
}
</style>
</head>
<body><div></div></body>
</html>
```

Esimerkkikoodi 2. Yksinkertainen CSS3-animaatio.

Toinen tapa, jota tässäkin työssä käytettiin, on toteuttaa CSS3-animaatio käyttämällä niin sanottuja sprite-arkkeja. Sprite-termillä tarkoitetaan irrallista kuvaa tai animaatiota, joka liitetään erillisen taustan päälle. Esimerkkinä tästä voisi olla vanhassa videopelissä animoitu hahmo, jota pystyy liikuttamaan itsenäisesti vaikuttamatta taustagrafiikkaan. Animaatiotekniikkana spriten käyttö itsessään on vanhaa. Sprite-arkki taas on yksittäinen iso arkki, joka koostuu useammasta pienestä kuvasta. Jokainen pieni kuva arkilla taas vastaa yhtä animaation vaihetta. (37.) Kuvassa 5 on esimerkki sprite-arkista, jota käytettiin työssä kohdassa, jossa paholaisen mökki ilmestyy savupilven saattelemana tyhjästä.



Kuva 5. 2004 x 135 pikselin kokoinen Sprite-arkki, jossa kuvattuna animaation vaiheet.

CSS3:lla animoitaessa sprite-arkkeja käytetään siten, että ensin määritellään animaation koko, siis tarkalleen ottaen alueen koko sivulla, jossa animaatio tapahtuu. Esimerkiksi kuvan 5 tapauksessa se on 143 x 135 pikseliä. Tämän jälkeen määritellään animaation kesto ja se, monessako vaiheessa animaatio tehdään. Kuvassa 5 esimerkissä kesto on kaksi sekuntia, ja koju syntyy tyhjästä kolmentoista vaiheen kautta. Itse animaatio tehdään siten, että arkista asemoidaan CSS:n avulla näkyviin ainoastaan yksi 143 x 135 pikselin kokoinen alue kerrallaan. Periaatteeltaan toteutus on hyvin samankaltainen kuin filmiprojektori, joka näyttää tietyn määrän filmiruutuja sekunnissa. Tässä tapauksessa filminauhaa vastaa sprite-arkki ja projektorina toimii CSS:ssä määritelty ohje. Kuvan 5 esimerkin käyttämä CSS-koodi löytyy kokonaisuudessaan liitteen 1 sisältämästä CSS-tyylitiedostossa.

Työssä interaktiivisuus animaatiota hyväksi käyttäen toteutettiin niin, että lukijan on itse käynnistettävä animaatio. Animaation käynnistys tehtiin yksinkertaisesti siten, että lukija klikkaa näytöllä vilkkuvaa, nappina toimivaa pulloa, joka sitten avaa seuraavan, muutoin samannäköisen HTML-sivun, mutta siinä animaatio lähtee automaattisesti käyntiin.

3.3.3 Ääni

Ennen HTML5-versioon siirtymistä ei HTML-merkintäkielessä ollut minkäänlaista standardia, miten ääntä toistetaan verkkosivulla, vaan äänet ja musiikki piti toistaa ulkopuolisilla liitännäisillä, kuten esimerkiksi Adoben Flash Playerillä. HTML5 toi uutena elementtinä <audio>-elementin, jonka tarkoituksena oli standardisoida äänen ja musiikin toisto verkkosivuilla (38). Interaktiivisessa sarjakuvassa voidaan käyttää valmiina olevia ääniefektejä ja musiikkia, mutta tässä työssä lähdettiin kuitenkin siitä, että kaikki työvaiheet, myös äänten ja musiikin äänitys, toteutetaan itse.

Ääniefektien ja musiikin äänittämiseen käytetty laitteisto ja äänitystavat voivat muun muassa projektin koosta, halutusta äänenlaadusta ja projektia tekevästä ryhmästä riippuen vaihdella suurestikin. Tässä nimenomaisessa projektissa äänitys oli vain yksi

pieni osa projektia, eikä kaikkia mahdollisia tapoja kannata tässä tarkemmin lähteä selostamaan, mutta kuten jo luvussa 3.2 mainittiin, toteutettiin äänitys tässä projektissa tavallisella, videota ja ääntä tallentavalla digikameralla. Sillä tallennettu ääni ei ehkä äänenlaadultaan ole huippuluokkaa, mutta se sopii riittävän hyvin 1930-luvulle, jolloin äänitystekniikka oli vielä suhteellisen alkeellista, sijoittuvaan sarjakuvaan.

Äänityslaitteistosta ja -tavasta riippumatta äänet tarvitsevat äänityksen jälkeen useimmiten vielä muokkausta. Muokkaus tehdään tietokoneella, johon siirretyt ääniraidat avataan DAW-ohjelmalla (Digital Audio Workstation eli digitaalisen äänen työasema). DAW:ssa ääniraidat editoidaan halutun pituisiksi eli poistetaan muun muassa ylimääräiset alku- ja loppukohinat. Tässä työssä muutamasta ääniraidasta haluttiin saada itseään katkeamatta toistava, joten niiden kohdalla piti katsoa, että raidan käynnistyessä uudelleen raidan loppu sopi riittävän saumattomasti raidan alkuun. Ääniraitojen äänenvoimakkuustasot on hyvä tässä vaiheessa myös säätää kohdalleen.

Valmiiksi muokatut ääniraidat tallennettiin ensin WAV-muodossa. WAV-muotoiset tiedostot ovat raakadataa, joten ne ovat suhteellisen isokokoisia, ja siksi ne piti vielä pakata, jotta niiden latautuminen itse sarjakuvassa on nopeampaa. Äänitiedostojen pakkauksessa tulee ottaa huomioon eri selainten tukemat tiedostomuodot. Neljä viidestä tämän hetken käytetyimmistä verkkoselaimista tukee MP3-muotoa (MPEG-1 Audio Layer 3). Kuten taulukosta 1 käy ilmi, nämä neljä selainta ovat Internet Explorer, Firefox, Safari ja Chrome. Opera-selain ei ainakaan toistaiseksi vielä tue MP3-muotoa, joten tiedostot on tarpeen pakata myös sen tukemaan Ogg-muotoon. (38.)

Taulukko 1. Yleisimpien selainten tukemat äänitiedostojen muodot.

Selain	WAV	MP3	Ogg
Firefox	tukee	tukee	tukee
Internet Explorer	ei tue	tukee	ei tue
Chrome	tukee	tukee	tukee
Safari	tukee	tukee	ei tue
Opera	tukee	ei tue	tukee

Äänten ja musiikin käyttö internetsarjakuvassa ei sellaisenaan tietenkään tee sarjakuvasta interaktiivista. Esimerkiksi jatkuvasti taustalla soiva musiikki, johon lukija ei pysty mitenkään vaikuttamaan, ei ole vuorovaikutteista, vaan lukijan pitää pystyä vaikuttamaan äänten kuulumiseen. Helppo tapa antaa lukijalle keino vuorovaikuttaa ääniin on lisätä sivulle toisto- ja pysäytysnappi, äänenvoimakkuuden säätö ja aikajana, jossa voi siirtyä ääniraidassa mihin kohtaan tahansa. Niiden lisääminen hoituu helposti HTML5:n <audio>-elementillä, jonka sisään tarvitsee vain lisätä controls-attribuutti. (39.)

Toinen tapa, jolla ääntä voidaan hyödyntää vuorovaikutuksen luomisessa, on sellainen, että lukijan viedessä hiiren osoittimen tiettyyn kohtaan sarjakuvan ruudussa kuuluu ääni. Tämän voi toteuttaa käyttämällä JavaScriptiä ja HTML5:n <audio>-elementtiä. Tällä kertaa <audio>-elementtiin ei tarvita controls-attribuuttia, vaan ainoastaan id- eli nimiattribuutti, jotta sitä pystytään JavaScript-funktiolla käyttämään. <audio>-elementin

lisäksi tarvitaan myös id-attribuutilla varustettu elementti, jonka päälle hiiren osoitin viettäessä ääni kuuluu. JavaScriptissä kerätään ensin molemmat mainitut elementit muuttujiin, minkä jälkeen määritetään, että onmouseover-tapahtuman (eli hiiren osoitin tulee elementin päälle) sattuessa toistetaan äänitiedosto. (40.) Esimerkkikoodissa 3 näkyvät edellä mainitut elementit ja funktiot.

```
<!--javascript-funktiot-->
<script>
//kerätään tiedot muuttujiin
    window.onload=function(){
        var guitar = document.getElementById('guitar');
        var playing = document.getElementById('playing');
/* toistetaan guitar-niminen äänitiedosto hiiren osoittimen
käydessä elementin päällä */
        playing.onmouseover=function(){
            guitar.play();
            return false; }; };
</script>
</head>
<body>
<div id="wrapper">
<a href="../../../inssi/index"></a>
<a href="../../../inssi/page10"><div id="back"></div></a>
<audio id="guitar" preload="auto">
<source src="sounds/f.mp3" type="audio/mpeg" />
<source src="sounds/f.ogg" type="audio/ogg" /></audio>
<!--elementti, jonka päälle osoitin viettäessä, äänitiedosto
toistetaan-->
<a id="playing" href="../../../inssi/index"></a>
```

Esimerkkikoodi 3. Äänen toisto hiiren osoittimen kulkiessa elementin päälle.

3.3.4 Video

Eräs keino toteuttaa interaktiivisuus on käyttää videoita. Insinööriyössä oli alun perin tarkoituksena saada käytettyä HTML5:n <video>-elementtiä jossakin kohdassa sarjakuvaa, mutta lopulta sille ei löytynyt mitään sopivaa kohtaa. Tähän saattoi osaltaan vaikuttaa menneisyyteen sijoittuva tarina, eli ei voinut esimerkiksi televisiota tai muuta näyttöä käyttää alustana videolle. Aivan kuten äänten ja musiikin toistamisenkin kanssa, ei videoiden toistamiselle internetsivuilla ollut minkäänlaista standardia ennen HTML5:sta, ja videot pitikin toistaa ulkopuolisilla mediasoittimilla, kuten Adoben Flash Playerillä (41).

HTML5:n <video>-elementti toimii hyvin samaan tapaan kuin luvussa 3.3.3 esitelty <audio>-elementti. Interaktiivisuus videoiden avulla voidaankin toteuttaa hyvin samaan tapaan kuin <audio>-elementtiä käyttämällä: lukijalle annetaan mahdollisuus toisto-, pysäytys-, kelaus- ja äänenvoimakkuudensäätönappien avulla vaikuttaa videoon. Ne saadaan käyttöön lisäämällä <video>-elementin sisään controls-attribuutti. (41.)

Toinen vaihtoehto on käynnistää video, kun lukija vie hiiren osoittimen sen päälle. Tämä voidaan toteuttaa lisäämällä <video>-elementin sisään onmouseover-attribuutti. Onmouseover-attribuutin arvoksi laitetaan tässä tapauksessa "this.play()" (eli toista), joka toistaa elementissä määritetyn videon. Haluttaessa lukijalle voidaan antaa myös mahdollisuus tauottaa videon toisto hiiren osoittimen poistuessa videon päältä lisäämällä elementin sisään onmouseout-attribuutti, jolle annetaan arvoksi "this.pause()" (eli tauko). Esimerkkikoodissa 4 on <video>-elementin sisään laitettu kummatkin attribuutit arvoineen.

```

<!DOCTYPE html>
<html>
<head>
<title>testi</title>
<meta charset="UTF-8">
</head>
<body>
<!--Videoelementti, johon on lisätty onmouseover- ja onmouseout-
attribuutit play- ja pause-arvoilla.-->
<video width="260"
onmouseover="this.play()" onmouseout="this.pause()">
  <source src="sounds/test.mp4" type="video/mp4">
  <source src="sounds/test.ogg" type="video/ogg">
  Your browser does not support HTML5 video.
</video>
</body>
</html>

```

Esimerkkikoodi 4. Videon toisto hiiren osoittimen kulkiessa elementin päälle.

HTML5:n <video>-elementtiä, aivan kuten <audio>-elementtiäkin, käyttäessä täytyy kuitenkin muistaa, että kaikki yleisimmät selaimet eivät tue samoja videotiedostoformaatteja. Varmin tapa, joskaan ei täysin varma, taata videoiden toimivuus sarjakuvassa on käyttää MP4-muotoa (MPEG-4 Part 14), jonka käyttöä kaikki viisi yleisintä selainta tukevat. Kaksi muuta selainten käytettävissä olevaa tiedostomuotoa ovat WebM ja Ogg. (41.) Taulukossa 2 on vielä listattuna, mitkä selaimista tukevat mitäkin tiedostomuotoja.

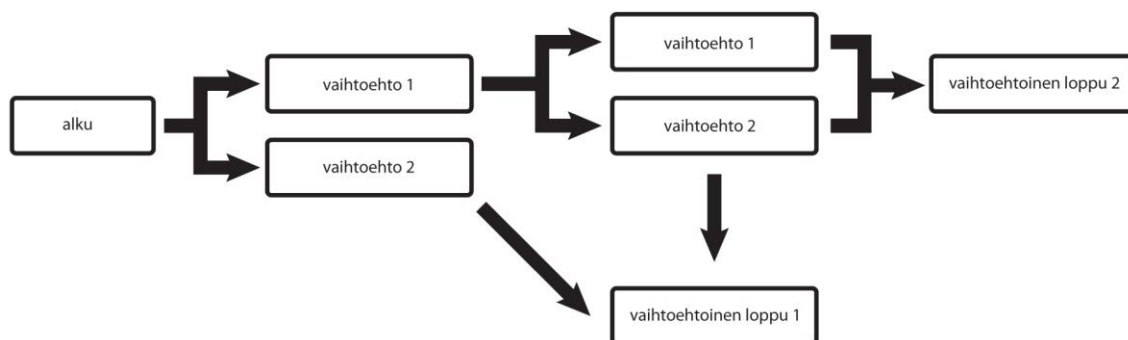
Taulukko 2. Yleisimpien selainten tukemat videotiedostojen muodot.

Selain	MP4	WebM	Ogg
Firefox	tukee	tukee	tukee
Internet Explorer	tukee	ei tue	ei tue
Chrome	tukee	tukee	tukee
Safari	tukee	ei tue	ei tue
Opera	tukee	tukee	tukee

3.4 Interaktiivisuuden vaikutus sarjakuvan juoneen

Yksi interaktiivisen ja perinteisen sarjakuvan suurimmista eroista on, että interaktiivisessa sarjakuvassa lukijalle voidaan antaa mahdollisuus eriasteisesti vaikuttaa sarjakuvan juoneen ja sen kulkuun. Tämä voi tarkoittaa valitsemista kahden eri asian välillä: esimerkiksi sarjakuvassa tulee vastaan kaksi ovea ja tarina etenee sen mukaan, kumman oven lukija valitsee.

Tämä vaihtoehtojen tarjoaminen vaatii sarjakuvan tekijältä paljon ylimääräistä suunnittelua ja työtä, sillä hänen täytyy suunnitella ja toteuttaa kaikki juonen mahdolliset variaatiot etukäteen, vaikka sarjakuvan lukija ei valitsekaan kuin tietyn reitin sarjakuvan läpi lukiessaan. Kuvassa 6 on esitetty yksinkertaistettuna insinööriyön juoni. Kaaviosta käyvät ilmi lukijan mahdollisuudet vaikuttaa juonen kulkuun.



Kuva 6. Sarjakuvan juoni yksinkertaistettuna.

Valintakohtien toteuttamiseen on useita eri keinoja. Sarjakuvan ensimmäinen juonihäärän valinta on sivulla 7, jossa lukijan pitää valita pakojuoksemisen ja paholaisen kioskillä vierailun välillä. Se toteutettiin yksinkertaisesti kahdella nappeina toimivalla kuvalla. Kun valitaan pakenemis-nappi, se linkkaa suoraan toiseen sarjakuvan kahdesta vaihtoehtoisesta lopusta, joka on esitetty kuvassa 6 nimellä vaihtoehtoinen loppu 1. Toinen nappi taas vie lukijan kohti seuraavaa valintakohtaa.

Tämä toinen, ja samalla viimeinen, valintakohta toteutettiin jo objektin liikuttamisesta kertovassa luvussa 3.3.1 läpikäydylä, objektin (tässä tapauksessa vastausvaihtoehdon) raahaus ja pudotus -tekniikalla HTML5:n drag-and-drop API:a hyväksi käyttäen. Sivulla on sopimuspaperi, jonka lukija joko hyväksyy tai hylkää raahaamalla ja pudottamalla valitsemansa vastausvaihtoehdon sille valitulle alueelle. Vastausvaihtoehdon irtipäästäminen aktivoi siinä mukana kulkevan linkin ja lukija siirtyy jompaankumpaan kahdesta vaihtoehtoisesta lopetussivusta.

4 Interaktiivisen sarjakuvan tulevaisuus

Tietyllä tapaa interaktiivisen sarjakuvan hyvänä puolena voidaan ainakin nähdä lukijalle annettu mahdollisuus vaikuttaa sarjakuvan kuluun. Myös tietokoneiden mahdollistamalla äänen ja videon käytöllä pystytään luomaan ja korostamaan tunnelmaa paremmin kuin pelkillä staattisilla kuvilla. Useat eri juoneenvaikeuttamiskohdat taas lisäävät sarjakuvan uudelleenlukemiskertoja ja tekevät uudelleenlukemisesta tietyllä tapaa mielekkäämpää, kun lukija ei ennalta tiedä, miten juoni uusilla valinnoilla etenee.

Yhtenä haittana voidaan nähdä interaktiivisen sarjakuvan kustannukset ja sen tekemiseen vaadittava aika. Toisena haittana voidaan nähdä interaktiivisten toiminnallisuuksien, koodauksen, mahdollisten juonihaarojen ynnä muiden suunnittelemisen ja luomisen vaatima oma erityisosaamisensa, kun taas perinteisen sarjakuvan tekoon riittää vähempi suunnittelu. Vertailuesimerkkinä voisi mainita vaikkapa kolmiruutuisen mustavalkoisen sarjakuvastripin verrattuna juonenkäänteitä sisältävään interaktiiviseen internetsarjakuvaan.

Ainakin toistaiseksi interaktiivisen sarjakuvan, kuten niin monen muunkin kokeellisen sarjakuvamuodon, suosio vaikuttaa jääneen pieneksi. Tämä kävi hyvin ilmi vähäisinä hakutuloksina insinööriöprojektin tiedonhakuvaiheessa. Koska interaktiivinen sarjakuva on sarjakuvan muotona vielä suhteellisen uusi, ei alan kirjallisuudessakaan siitä juuri tietoa ollut, eikä internetin hakukoneillakaan tullut kovin paljon osumia. Myös itse interaktiivisten sarjakuvien löytyminen internetistä oli todella vähäistä.

Ehkäpä suurimpana syynä tähän vähäiseen suosioon näkisin sen, että interaktiivinen sarjakuva jää kellumaan perinteisen sarjakuvan, animaation ja ehkä jopa tietyllä tapaa tietynlaisten tietokonepelien välimaastoon. Otetaan esimerkkinä kuvitteellinen interaktiivinen sarjakuva, jossa lukija pystyy esimerkiksi liikuttamaan hahmoa sarjakuvan ruuduilla ja jossa lukijalla on mahdollisuus vaikuttaa juoneen. Sehän on periaatteessa jo tietyn tyyppinen yksinkertainen tietokonepeli. Ehkäpä tästä syystä perinteisen sarjakuvan lukijat, eivätkä toisaalta myöskään animaation tai tietokonepelien ystävät, pidä sitä omanaan. Juuri tämän välimuotoisuuden vuoksi minun on ainakin itse vaikea kuvitella sen suosion juurikaan nousevan.

Toisena syynä vähäiseen suosioon saattaa yksinkertaisesti olla osaavien ja halukkaiden tekijöiden vähäinen määrä. Interaktiivisen sarjakuvan toteuttamiseen vaatima erityisosaaminen, muun muassa koodaus ja äänten ja videoiden tuottaminen sekä niistä johtuva työmäärä, varmasti karsii halukkaiden tekijöiden määrää.

5 Yhteenveto

Insinööriöprojekti aloitettiin tutustumalla sarjakuvan historiaan ja siihen, miten se on kehittynyt digitaalisiin muotoihinsa ja lopulta siirtynyt internetiin. Tämän jälkeen tutustuttiin sarjakuvan tekemisen eri työvaiheisiin ja niissä käytettäviin työvälineisiin.

Tiedonhakuvaiheiden jälkeen aloitettiin sarjakuvan suunnittelu, jonka aikana otettiin jo huomioon sarjakuvaan tulevat interaktiiviset kohdat. Suunnittelua seurasi itse ruutujen luonnostelu, piirtäminen ja puhtaaksi piirtäminen, minkä jälkeen tehtiin sarjakuvan vaatima koodaus. Koodaus tehtiin käyttämällä HTML5-merkintäkieltä ja siihen liittyviä teknologioita, kuten CSS3:a ja Javascript-kirjastoja.

Insinööri työ onnistui asetettuihin tavoitteisiin nähden hyvin. Sarjakuvan historia ja sen kehitys, sarjakuvanteon eri työvaiheet ja työvälit tulivat projektin myötä tutuiksi. Sarjakuvan historiaan tutustuminen osoitti, että vaikka sarjakuvallisella kerronnalla on pitkät perinteet, sai nykyaikainen sarjakuva alkunsa vasta 1800-luvun loppupuoliskolla. Kävi myös selväksi, että teknologian kehityksestä ja sarjakuvan saamista uusista muodoista huolimatta sarjakuva on säilynyt periaatteiltaan hyvin samanlaisena aina tähän päivään asti. Työn lopputuloksena oli ehkä hieman yksinkertainen, mutta kuitenkin kaikki halutut interaktiiviset ominaisuudet, muun muassa juoneen vaikuttamismahdollisuuksien ja objektien liikuttamisen, sisältävä interaktiivinen sarjakuva. Eikä myöskään mitään suurempia ongelmia ilmennyt valittujen toiminnallisuuden toteuttamisessa.

Projekti osoitti, että interaktiivisuus voidaan toteuttaa internetsarjakuvaan useita eri alustoja, keinoja ja menetelmiä käyttämällä. Tavoitteeksikin listattu mahdollisimman suuri HTML5:n käyttö laajensi HTML5:n osaamista muun muassa <audio>- ja <video>-elementtien sekä drag-and-drop API:n osalta. Projektin myötä uutena asiana tuli myös oppia, miten CSS3:n avulla voidaan toteuttaa animaatioita. Myös entuudestaan tuntemattomien ja käyttämättömien JavaScript-kirjastojen, jQuery:n ja jQuery UI:n, käyttö yhdessä HTML5:n kanssa tuli kohtalaisen tutuksi. Opin myös, että ennakkoletuksesta poiketen interaktiivisuuden toteuttaminen ei vaadi mitenkään monimutkaista koodaustaitoa.

Projektin myötä kävi myös selväksi, että interaktiivinen sarjakuva on vielä marginaalinen ilmiö sarjakuvien maailmassa, ja kovasti vaikuttaa siltä, että se myös jää sellaiseksi.

Lähteet

- 1 McCloud, Scott. 2000. Reinventing Comics. New York: HarperCollins Publishers.
- 2 Waugh, Coulton. 2010. The Comics. Jackson: University Press of Mississippi.
- 3 Jokinen, Heikki (toim.). 2011. Sarjakuva Suomessa. Helsinki: BTJ Finland.
- 4 VanderPloeg, Scott. 2012. The Importance of Panel Layout. Verkkodokumentti. Comic Book Daily. <<http://www.comicbookdaily.com/columns/jiminy-christmas/the-importance-of-panel-layout/>>. Luettu 19.9.2013.
- 5 Coville, Jamie. 1996. The Platinum Age 1897 - 1938. Verkkodokumentti. <<http://www.thecomicbooks.com/old/Platinum.html>>. Luettu 20.9.2013.
- 6 Coville, Jamie. 1996. The Golden Age (1938 - 1956). Verkkodokumentti. <<http://www.thecomicbooks.com/old/Hist1.html>>. Luettu 20.9.2013.
- 7 Farr, Michael. 2011. Tintti tarinoiden todelliset taustat. Helsinki: Kustannusosakeyhtiö Otava.
- 8 The Golden Age Of Comics. Verkkodokumentti. Public Broadcasting Service. <<http://www.pbs.org/opb/historydetectives/feature/the-golden-age-of-comics/>>. Luettu 11.11.2013.
- 9 What is Digital Drawing? Verkkodokumentti. Creative Comic Art. <<http://www.creativecomicart.com/digital-drawing.html>>. Luettu 16.3.2015.
- 10 Batman: Digital Justice. Verkkodokumentti. Comic Vine. <<http://www.comicvine.com/batman-digital-justice/4050-28204/>>. Luettu 13.11.2013.
- 11 Lassila, Ilkka. 2011. Sarjakuvan uudet muodot. Verkkodokumentti. Opinnäytetyö. Oulun seudun ammattikorkeakoulu. <http://publications.theseus.fi/bitstream/handle/10024/27356/Lassila_Ilkka.pdf>. Luettu 20.9.2013.
- 12 Smith, K. Alexander. 2011. 14 Awesome Webcomics To Distract You From Getting Things Done. Verkkodokumentti. <<http://www.pastemagazine.com/blogs/lists/2011/09/22-awesome-webcomics.html>>. Luettu 21.11.2013.
- 13 Kean, Benjamin. 2008. Economics and Comics: How Do Webcomics Fit In? Verkkodokumentti. <<http://www.newsarama.com/150-economics-and-comics-how-do-webcomics-fit-in.html>>. Luettu 22.11.2013.

- 14 Cohen, Noam. 2008. This Is Funny Only if You Know Unix. Verkkodokumentti. New York Times.
<http://www.nytimes.com/2008/05/26/business/media/26link.html?partner=rssnyt&emc=rss&_r=0>. Luettu 20.6.2014.
- 15 Fingerpori. Verkkodokumentti. Helsingin Sanomat. <<http://www.hs.fi/fingerpori/>>. Luettu 30.3.2015.
- 16 Munroe, Randall. Xkcd. Verkkodokumentti. <<http://xkcd.com/570/>>. Luettu 21.9.2013.
- 17 Interaktiivisuus. Verkkodokumentti. Suomen mediaopas.
<<http://www.mediaopas.com/sanasto/interaktiivisuus>>. Luettu 25.9.2013.
- 18 Yllyke. Verkkodokumentti. Sarjakuvantekowiki.
<<http://www.sarjakuvanteko.fi/wiki/index.php?title=Yllyke>>. Luettu. 13.6.2014.
- 19 Sarjakuvan tekemisen vaiheet. Verkkodokumentti. Sarjakuvantekowiki.
<<http://www.sarjakuvanteko.fi/wiki/index.php?title=Etusivu>>. Luettu 13.6.2014.
- 20 Suunnitteleminen. Verkkodokumentti. Sarjakuvantekowiki.
<<http://www.sarjakuvanteko.fi/wiki/index.php?title=Suunnitteleminen>>. Luettu 12.3.2014.
- 21 Tekstikäsikirjoitus. Verkkodokumentti. Sarjakuvantekowiki.
<<http://www.sarjakuvanteko.fi/wiki/index.php?title=Tekstik%C3%A4sikirjoitus>>. Luettu 6.6.2014.
- 22 Tommy Johnson. Verkkodokumentti. National Park Service.
<http://www.nps.gov/history/delta/blues/people/tommy_johnson.htm>. Luettu 20.4.2014.
- 23 Robert Johnson. Verkkodokumentti. National Park Service.
<http://www.nps.gov/history/delta/blues/people/robert_johnson.htm>. Luettu 20.4.2014.
- 24 Sketch. Verkkodokumentti. Encyclopædia Britannica.
<<http://global.britannica.com/EBchecked/topic/547463/sketch>>. Luettu 13.4.2015.
- 25 Kuvakäsikirjoitus. Verkkodokumentti. Sarjakuvantekowiki.
<<http://www.sarjakuvanteko.fi/wiki/index.php?title=Kuvak%C3%A4sikirjoitus>>. Luettu 22.7.2014.
- 26 Puhtaaksi piirtäminen. Verkkodokumentti. Sarjakuvantekowiki.
<http://www.sarjakuvanteko.fi/wiki/index.php?title=Puhtaaksi_piirt%C3%A4minen>. Luettu 22.7.2014.

- 27 Standard Resolutions. Verkkodokumentti. #quasys GmbH.
<<http://www.equasys.de/standardresolution.html>>. Luettu 12.4.2015.
- 28 Väritys. Verkkodokumentti. Suomen Sarjakuvaseura.
<<http://sarjakuvaseura.fi/akatemia/frameset.html?akatemia/kerros3/luokka3a/vvaihdeet.html>>. Luettu 13.4.2015.
- 29 Avoimen lähdekoodin määritelmä. Verkkodokumentti. EDU.fi.
<http://www.edu.fi/valo_opas/avoin_lahdekoodi_maaritelma>. Luettu 22.7.2014.
- 30 JQuery introduction. Verkkodokumentti. W3Schools.
<http://www.w3schools.com/jquery/jquery_intro.asp>. Luettu 12.5.2015.
- 31 CSS-opas. Verkkodokumentti. KK Mediat.
<<http://www.2kmediat.com/css/johdanto.asp>>. Luettu 22.7.2014.
- 32 JQuery UI Draggable. Verkkodokumentti. The jQuery Foundation.
<<http://jqueryui.com/draggable/>>. Luettu 21.7.2014.
- 33 HTML5 Drag and Drop. Verkkodokumentti. W3Schools.
<http://www.w3schools.com/html/html5_draganddrop.asp>. Luettu 20.7.2014.
- 34 What is Edge Animate? Verkkodokumentti. Adobe.
<<https://helpx.adobe.com/creative-cloud/tutorials/videos/what-is-edge-animate.html?product=animate&path=get-started>>. Luettu 25.11.2014.
- 35 Mikä Google Web Designer on? Verkkodokumentti. Google.
<https://support.google.com/webdesigner/answer/3184833?hl=fi&ref_topic=3249465>. Luettu 25.11.2014.
- 36 CSS3 Animation. Verkkodokumentti. W3Schools.
<http://www.w3schools.com/css/css3_animations.asp>. Luettu 20.7.2014.
- 37 Lambert, Steven. 2013. An Introduction to Spritesheet Animation. Verkkodokumentti. <<http://gamedevelopment.tutsplus.com/tutorials/an-introduction-to-spritesheet-animation--gamedev-13099>>. Luettu 13.4.2015.
- 38 HTML5 Audio. Verkkodokumentti. W3Schools.
<http://www.w3schools.com/html/html5_audio.asp>. Luettu 20.7.2014.
- 39 Granell G., Sumner V. & Synodinos D. 2012. The Essential Guide To HTML5 and CSS3 Web Design. USA. Friends of ED.

- 40 Aravind Buddha. 2012. Playing Sounds On Mouse Hover Using HTML5. Verkkodokumentti. Techumber.com.
<<http://www.techumber.com/2012/12/playing-sounds-on-mouse-hover-using-HTML5.html>>. Luettu 11.4.2015.
- 41 HTML5 Video. Verkkodokumentti. W3Schools.
<http://www.w3schools.com/html/html5_video.asp>. Luettu 20.7.2014.

Lähdekoodi ja CSS-tyylitiedosto

Sisältää lähdekoodin sekä CSS-tyylitiedoston.

<!-- kansilehti -->

<!DOCTYPE html>

<html>

<head>

<title>Crossroads</title>

<meta charset="UTF-8">

<link rel="stylesheet" type="text/css" href="css/leiska.css">

</head>

<body>

<div id="wrapper">

<audio controls autoplay loop>

<source src="sounds/introfades.mp3" type="audio/mpeg" />

<source src="sounds/introfades.ogg" type="audio/ogg" /></audio>

</div>

<div id="footer"></div>

</body>

</html>

<!-- 1. sivu -->

<!DOCTYPE html>

<html>

<head>

<title>Crossroads</title>

<meta charset="UTF-8">

<link rel="stylesheet" type="text/css" href="css/leiska.css">

</head>

<body>

<div id="wrapper">

```

<div id="eyes"></div>
<a href="../inssi/index"><div id="back"></div></a>
<a href="../inssi/page2"><div id="button"></div></a>
</div>
<div id="footer"></div>
</body>
</html>
```

<!-- 2. sivu -->

```
<!DOCTYPE html>
<html>
<head>
<title>Crossroads</title>
<meta charset="UTF-8">
<link rel="stylesheet" type="text/css" href="css/leiska.css">
<script src="js/jquery-1.11.1.min.js"></script>
<script src="js/jquery-ui.min.js"></script>
<script>
$(document).ready(function() {
    $('#bottlebutton1').delay(3000).fadeIn(400);
});
</script>
</head>
<body>
<div id="wrapper">
<audio autoplay loop>
<source src="sounds/outoftune.mp3" type="audio/mpeg" />
<source src="sounds/outoftune.ogg" type="audio/ogg" /></audio>

<a href="../inssi/page1"><div id="back"></div></a>
<div class="bottle"></div>
<div id="bottlebutton"><a href="../inssi/page2,5"></a></div>
```

```
</div>  
<div id="footer"></div>  
</body>  
</html>
```

<!-- 2½. sivu ->

```
<!DOCTYPE html>  
<html>  
<head>  
<title>Crossroads</title>  
<meta charset="UTF-8">  
<link rel="stylesheet" type="text/css" href="css/leiska.css">  
</head>  
<body>  
<div id="wrapper">  
  
<a href="../inssi/page2"><div id="back"></div></a>  
<div class="throw"></div>  
<a href="../inssi/page3"><div id="button"></div></a>  
</div>  
<div id="footer"></div>  
</body>  
</html>
```

<!-- 3. sivu ->

```
<!DOCTYPE html>  
<html>  
<head>  
<title>Crossroads</title>  
<meta charset="UTF-8">  
<link rel="stylesheet" type="text/css" href="css/leiska.css">  
</head>  
<body>
```

```
<div id="wrapper">

<a href="../inssi/page2,5"><div id="back"></div></a>
<a href="../inssi/page4"><div id="button"></div></a>
</div>
<div id="footer"></div>
</body>
</html>
```

<!-- 4. sivu -->

```
<!DOCTYPE html>
<html>
<head>
<title>Crossroads</title>
<meta charset="UTF-8">
<link rel="stylesheet" type="text/css" href="css/leiska.css">
<script src="js/jquery-1.11.1.min.js"></script>
<script src="js/jquery-ui.min.js"></script>
<script>
$(function() {
$( '#draggable' ).draggable({ containment: "parent"});
});
</script>
</head>
<body>
<div id="wrapper">

<a href="../inssi/page3"><div id="back"></div></a>
<a href="../inssi/page5"><div id="button"></div></a>
<div id="dragarea">
<div id="draggable"></div>
</div>
</div>
<div id="footer"></div>
```

```
</body>  
</html>
```

<!-- 5. sivu -->

```
<!DOCTYPE html>  
<html>  
<head>  
<title>Crossroads</title>  
<meta charset="UTF-8">  
<link rel="stylesheet" type="text/css" href="css/leiska.css">  
</head>  
<body>  
<div id="wrapper">  
  
<a href="../inssi/page4"><div id="back"></div></a>  
<a href="../inssi/page6"><div id="button"></div></a>  
</div>  
<div id="footer"></div>  
</body>  
</html>
```

<!-- 6. sivu -->

```
<!DOCTYPE html>  
<html>  
<head>  
<title>Crossroads</title>  
<meta charset="UTF-8">  
<link rel="stylesheet" type="text/css" href="css/leiska.css">  
</head>  
<body>  
<div id="wrapper">  

```



```
<a href="../inssi/page5"><div id="back"></div></a>
<a href="../inssi/page7"><div id="button"></div></a>
</div>
<div id="footer"></div>
</body>
</html>
```

<!-- 7. sivu -->

```
<!DOCTYPE html>
<html>
<head>
<title>Crossroads</title>
<meta charset="UTF-8">
<link rel="stylesheet" type="text/css" href="css/leiska.css">
<script src="js/jquery-1.11.1.min.js"></script>
<script src="js/jquery-ui.min.js"></script>
<script>
$(document).ready(function() {
    $('#run2').delay(5000).fadeIn(400);
    $('#deals2').delay(5000).fadeIn(400);
});
$(document).ready(function(){
    $('#run2').hover(function() {
        $(this).attr("src","images/run2.png");
    }, function() {
        $(this).attr("src","images/run.png");
    });
    $('#deals2').hover(function() {
        $(this).attr("src","images/deals2.png");
    }, function() {
        $(this).attr("src","images/deals.png");
    });
});
</script>
```

```
</head>
<body>
<div id="wrapper">

<a href="../inssi/page6"><div id="back"></div></a>
<div class="hut"></div>
<div class="huh"></div>
<div id="run"><a href="../inssi/page8"></a></div>
<div id="deals"><a href="../inssi/page9"></a></div>
</div>
<div id="footer"></div>
</body>
</html>
```

<!-- 8. sivu -->

```
<!DOCTYPE html>
<html>
<head>
<title>Crossroads</title>
<meta charset="UTF-8">
<link rel="stylesheet" type="text/css" href="css/leiska.css">
<script>
  window.onload=function(){
    var guitar = document.getElementById('guitar');
    var playing=document.getElementById('playing');
    playing.onmouseover=function(){
      guitar.play();
      return false;
    };
  };
</script>
</head>
```

```
<body>
<div id="wrapper">
<a href="../inssi/index"></a>
<a href="../inssi/page7"><div id="back"></div></a>
<audio id="guitar" preload="auto">
<source src="sounds/f.mp3" type="audio/mpeg" />
<source src="sounds/f.ogg" type="audio/ogg" /></audio>
<a id="playing" href="../inssi/index"></a>
</div>
<div id="footer"></div>
</body>
</html>
```

<!-- 9. sivu -->

```
<!DOCTYPE html>
<html>
<head>
<title>Crossroads</title>
<meta charset="UTF-8">
<link rel="stylesheet" type="text/css" href="css/leiska.css">
</head>
<body>
<div id="wrapper">

<a href="../inssi/page7"><div id="back"></div></a>
<a href="../inssi/page10"><div id="button"></div></a>
</div>
<div id="footer"></div>
</body>
</html>
```

<!-- 10. sivu -->

```
<!DOCTYPE html>
<html>
<head>
<title>Crossroads</title>
<meta charset="UTF-8">
<link rel="stylesheet" type="text/css" href="css/leiska.css">
<script>
function allowDrop(ev)
    {
        ev.preventDefault();
    }
function drag(ev)
    {
        ev.dataTransfer.setData("text", "../inssi/page11");
    }
function dragTwo(ev)
    {
        ev.dataTransfer.setData("text", "../inssi/page12");
    }
function drop(ev)
    {
        ev.preventDefault();
        window.location=ev.dataTransfer.getData("text");
    }
</script>
</head>
<body>
<div id="wrapper">

<a href="../inssi/page9"><div id="back"></div></a>
<div id="dropzone" ondrop="drop(event)" ondragover="allowDrop(event)"></div>


```

```
</div>  
<div id="footer">  
</div>  
</body>  
</html>
```

<!-- 11. sivu -->

```
<!DOCTYPE html>  
<html>  
<head>  
<title>Crossroads</title>  
<meta charset="UTF-8">  
<link rel="stylesheet" type="text/css" href="css/leiska.css">  
</head>  
<body>  
<div id="wrapper">  
<audio autoplay>  
<source src="sounds/outro.mp3" type="audio/mpeg" />  
<source src="sounds/outro.ogg" type="audio/ogg" /></audio>  
<a href="../inssi/index.html"></a>  
<a href="../inssi/page10"><div id="back"></div></a>  
</div>  
<div id="footer"></div>  
</body>  
</html>
```

<!-- 12. sivu -->

```
<!DOCTYPE html>  
<html>  
<head>  
<title>Crossroads</title>  
<meta charset="UTF-8">
```

```
<link rel="stylesheet" type="text/css" href="css/leiska.css">
<script>
  window.onload=function(){
    var guitar = document.getElementById('guitar');
    var playing=document.getElementById('playing');
    playing.onmouseover=function(){
      guitar.play();
      return false;
    };
  };
</script>
</head>
<body>
<div id="wrapper">
<a href="../inssi/index"></a>
<a href="../inssi/page10"><div id="back"></div></a>
<audio id="guitar" preload="auto">
<source src="sounds/f.mp3" type="audio/mpeg" />
<source src="sounds/f.ogg" type="audio/ogg" /></audio>
<a id="playing" href="../inssi/index"></a>
</div>
<div id="footer"></div>
</body>
</html>
```

CSS-tyylitiedosto

```
html {  
  height: 100%;  
}  
body {  
  margin: -3px;  
  background-color: #535353;  
  background-repeat: repeat;  
  overflow-y: scroll;  
  overflow-x: hidden;  
}  
a {  
  outline: none;  
  text-decoration: none;  
}  
img {  
  max-width: 100%;  
}  
audio {  
  width: 808px;  
  height: 25px;  
}  
#wrapper {  
  width: 808px;  
  height: 691px;  
  margin: 3px auto 0 auto;  
  border-top-left-radius: 0.7em;  
  border-top-right-radius: 0.7em;  
  box-shadow: 0 -15px 20px #000;  
  background-color: #ffffff;  
  min-height: 100%;  
}  
.hut {  
  background-image: url("../images/hut1.png");
```

```
background-repeat: no-repeat;
width: 143px;
height: 135px;
margin-left: 142px;
margin-top: -330px;
float: left;
position: relative;
z-index: 1;
animation: hut 2s steps(13);
animation-delay: 3s;
animation-iteration-count: 1;
animation-fill-mode: forwards;
-webkit-animation: hut 2s steps(13);
-webkit-animation-delay: 3s;
-webkit-animation-iteration-count: 1;
-webkit-animation-fill-mode: forwards;
}
@keyframes hut {
from {background-position: 0 0;}
to {background-position: -1861px 0;}
}
@-webkit-keyframes hut {
from {background-position: 0 0;}
to {background-position: -1861px 0;}
}
.huh {
background-image: url("../images/huh.png");
background-repeat: no-repeat;
width: 72px;
height: 65px;
margin-right: 200px;
margin-top: -330px;
float: right;
position: relative;
z-index: 1;
```



```
animation: huh 2s steps(1);
animation-delay: 2s;
animation-iteration-count: 1;
animation-fill-mode: forwards;
-webkit-animation: huh 2s steps(1);
-webkit-animation-delay: 2s;
-webkit-animation-iteration-count: 1;
-webkit-animation-fill-mode: forwards;
}

@keyframes huh {
from {background-position: 0 0;}
to {background-position: -72px 0;}
}

@-webkit-keyframes huh {
from {background-position: 0 0;}
to {background-position: -72px 0;}
}

.bottle {
animation: flash 2s steps(2) infinite;
-webkit-animation: flash 2s steps(2) infinite;
background-image: url("../images/flash.png");
background-repeat : no-repeat;
width: 808px;
height: 250px;
margin-left: 215px;
margin-top: -268px;
float: left;
position: relative;
z-index: 1;
}

@keyframes flash {
0% {background-position: 0 0;}
100% {background-position: 0 -450px;}
}

@-webkit-keyframes flash {
```

```
0% {background-position: 0 0;}
100% {background-position: 0 -450px;}
}

.throw {
background-image: url("../images/throw1.png");
background-repeat: no-repeat;
width: 415px;
height: 400px;
margin-left: 152px;
margin-top: -515px;
float: left;
position: relative;
z-index: 1;
animation: throw 2s steps(15);
animation-delay: 3s;
animation-iteration-count: 1;
animation-fill-mode: forwards;
-webkit-animation: throw 2s steps(15);
-webkit-animation-delay: 3s;
-webkit-animation-iteration-count: 1;
-webkit-animation-fill-mode: forwards;
}

@keyframes throw {
from {background-position: 0px 0; background-image : url("../images/throw1.png");}
to {background-position: -6227px 0;}
}

@-webkit-keyframes throw {
from {background-position: 0px 0; background-image : url("../images/throw1.png");}
to {background-position: -6227px 0;}
}

#back {
background-image: url("../images/nuoli3.png");
width: 75px;
height: 29px;
float: left;
```

```
position: relative;
margin-left: 10px;
margin-top: -70px;
z-index: 1;
}
#button {
background-image: url("../images/nuoli.png");
width: 75px;
height: 29px;
float: right;
position: relative;
margin-right: 77px;
margin-top: -80px;
z-index: 1;
}
#bottlebutton {
width: 45px;
height: 152px;
position: relative;
margin-left: 233px;
margin-top: -260px;
z-index: 1;
}
#bottlebutton1 {
display: none;
width: 45px;
height: 152px;
margin-top: -2px;
}
#playing {
background-image: url("../images/dude.png");
width: 280px;
height: 262px;
float: right;
position: relative;
```

```
margin-right: 152px;
margin-top: -432px;
z-index: 1;
}
#sign {
background-image: url("../images/sign.png");
width: 200px;
height: 41px;
float: right;
position: relative;
margin-right: 110px;
margin-top: -100px;
z-index: 1;
}
#dontsign {
background-image: url("../images/dontsign.png");
width: 200px;
height: 39px;
float: right;
position: relative;
margin-top: -100px;
z-index: 2;
margin-right: 430px;
}
#deals2 {
display: none;
width: 200px;
height: 46px;
}
#run2 {
display: none;
width: 200px;
height: 52px;
}
#run {
```

```
width: 200px;
height: 52px;
float: left;
position: relative;
margin-top: -100px;
z-index: 2;
margin-left: 450px;
}
#deals {
width: 200px;
height: 46px;
float: right;
position: relative;
margin-top: -100px;
z-index: 2;
margin-right: 420px;
}
#eyes{
background-image: url("../images/eyes.png");
width: 24px;
height: 13px;
float: right;
position: relative;
margin-top: -194px;
z-index: 1;
margin-right: 468px;
}
#eyes:hover {
background-image: none;
}
#back:hover{
background-image: url("../images/nuoli4.png");
}
#button:hover{
background-image: url("../images/nuoli2.png");
```

```
}  
#draggable {  
width: 72px;  
height: 42px;  
background-image: url("../images/dragbottle.png");  
z-index: 1;  
}  
#dragarea{  
float: right;  
position: relative;  
margin-right: 110px;  
margin-top: -165px;  
z-index: 1;  
width: 390px;  
height: 80px;  
}  
#dropzone {  
float: right;  
position: relative;  
margin-right: 260px;  
margin-top: -285px;  
z-index: 1;  
width: 180px;  
height: 57px;  
}  
#drag1 {  
float: left;  
position: relative;  
margin-left: 170px;  
margin-top: -100px;  
z-index: 1;  
}  
#drag2 {  
float: right;  
position: relative;
```

```
margin-right: 90px;
margin-top: -100px;
z-index: 1;
}
#footer {
background-image: url("../images/footer.png");
background-repeat: no-repeat;
margin-bottom: 0;
position: relative;
height: 35px;
width: 808px;
margin-top: -73px;
clear: both;
margin: auto;
box-shadow: 0 0 20px #000;
z-index: -1;
}
```